

SWORD

XDI-SWORD-IMG-007

Lab7: 图像处理滤波器实验

4: 二值腐蚀膨胀

Joseph Xu

2018-5-13

修改记录

版本号.	作者	描述	修改日期
1.0	Joseph Xu	初稿	2018-5-13

审核记录

姓名	职务	签字	日期

目录

修改记录.....	1
审核记录.....	1
1. 实验简介.....	6
1.1 概述.....	6
1.2 实验目标.....	7
1.3 实验条件.....	7
1.4 实验原理.....	7
2. 二值图腐蚀实验流程.....	12
2.1 操作步骤.....	12
3. 腐蚀实验结果.....	31


	标题	文档编号	版本	页
	Lab7: 图像处理滤波器实验 4	XDI-SWORD-IMG-007	1.0	2 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/5/13		

图 1-1	实验连接示意图.....	8
图 1-2	图像对比度调节连接示意图.....	8
图 1-3	RowGenerator IP	9
图 1-4	WindowGenerator IP	9
图 1-5	MeanFilter IP	10
图 2-1	复制一个实验 6 副本.....	12
图 2-2	重命名实验目录.....	12
图 2-3	启动 Vivado.....	13
图 2-4	打开工程.....	13
图 2-5	实验初始视图.....	14
图 2-6	添加局部阈值化 IP.....	14
图 2-7	添加腐蚀膨胀 IP 后的视图.....	15
图 2-8	腐蚀膨胀 IP 配置.....	15
图 2-9	添加 RowsGenerator IP	16
图 2-10	添加 WindowGenerator IP	16
图 2-11	RowsGenerator IP 配置	17
图 2-12	WindowGenerator IP 配置	18
图 2-13	添加反色 IP	18
图 2-14	配置反色 IP	19
图 2-15	断开 ThresholdLocal IP 的 out_data 端口	19
图 2-16	断开端口后视图.....	20
图 2-17	连接 ColorReversal 和 RowsGenerator	21
图 2-18	连接 RowsGenerator 和 WindowGenerator	21
图 2-19	连接腐蚀膨胀 IP	22
图 2-20	添加常数 IP	22
图 2-21	添加 2 个常数 IP	23
图 2-22	配置腐蚀模式.....	23
图 2-23	配置腐蚀模式.....	24
图 2-24	端口连接检查.....	24
图 2-25	保存设计.....	25
图 2-26	创建实验顶层 Wrapper 文件	25
图 2-27	自动更新顶层文件.....	25




	标题	文档编号	版本	页
	Lab7: 图像处理滤波器实验 4	XDI-SWORD-IMG-007	1.0	3 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/5/13		

图 2-28	打开 lab1.xdc 文件	26
图 2-29	修改 lab1.xdc 文件	26
图 2-31	Generate Bitstream	26
图 2-32	点击 Yes 确认生成 bit 文件	27
图 2-33	打开 Hardware Manager	27
图 2-34	硬件连接对应位置	28
图 2-35	实际硬件连接	28
图 2-36	Open target	29
图 2-37	Program Device	29
图 2-38	烧写目标器件	30
图 2-39	编程进度条	30
图 3-1	局部阈值化显示结果	31

	标题	文档编号	版本	页
	Lab7: 图像处理滤波器实验 4	XDI-SWORD-IMG-007	1.0	4 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/5/13		

表目录

表 1	RowGenerator IP 端口列表	9
表 2	WindowGenerator IP 端口列表.....	10
表 3	MeanFilter IP 端口列表	10

	标题	文档编号	版本	页
	Lab7: 图像处理滤波器实验 4	XDI-SWORD-IMG-007	1.0	5 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/5/13		

1. 实验简介

该实验通过一定尺寸的滑动窗口对原始的二值化图像数据进行边界的缩小（腐蚀）或扩大（膨胀）计算。从而强化一定的特征，弱化一定的噪声。

- **对于初学者，整个实验预计耗时 1 小时。**
- **对于进阶者，整个实验预计耗时 10 分钟。**

1.1 概述

形态学滤波器是很常用的一种局部滤波器，顾名思义，它针对图像的形态进行操作，关注的是“形状”。形态学滤波一般分为腐蚀和膨胀，而在其之上又可以叠加为开运算和闭运算，它对二值图像、灰度图像均有效，但二者实现的方式不同，一般最常用的是二值形态学滤波，针对灰度的本质上是极大值和极小值排序滤波器。形态学滤波常用于图像细化、骨架提取等，可以作为图像识别的一种基本的预处理操作。

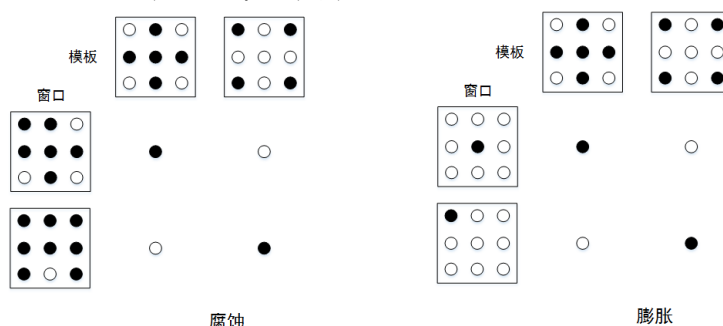
二值形态学滤波的操作单元与其他局部滤波器一致，都是包含了某一个中心像素点及其邻域的窗口，但比起其他操作，它还有一个作为参照的“模板”，并根据模板和窗口的运算来得到输出。所有的二值图像都可以分为主体像素和背景像素，一般以 1 为主体像素，0 为背景像素，这样最基本的二值形态学操作-腐蚀和膨胀便可以定义为式 1 和式 2：


$$\text{腐蚀: } Q = \bigvee_{i,j \in T} I[x+i, y+j] \quad (\text{式 1})$$

$$\text{膨胀: } Q = \bigwedge_{i,j \in T} I[x-i, y-j] \quad (\text{式 2})$$

其中 I 为输入， T 为模板， \vee 为逻辑或操作， \wedge 为逻辑与操作。

可见腐蚀是一种收缩的变换，它将窗口与模板进行对比，如果模板中每一个主体像素都落在窗口内，则此时窗口的中心像素置 1，否则为 0；而膨胀则是一个扩张的变换，只要模板中的任一主体像素落在窗口内，则中心像素置 1。腐蚀和膨胀的原理示意如下图所示，空心的点表示 0，否则表示 1。



	标题	文档编号	版本	页
	Lab7: 图像处理滤波器实验 4	XDI-SWORD-IMG-007	1.0	6 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/5/13		

1.2 实验目标

本实验的目标为 SWORD4.0 能够对 HDMI 输入的图像画面进行腐蚀或膨胀后在显示器上输出的视频画面。


1.3 实验条件

类别	名称	数量	说明
硬件	SWORD4.0	1	
	HDMI 信号源	1	如笔记本 HDMI 输出/台式计算机 HDMI 输出/带 HDMI 输出的视频机顶盒
	带 HDMI 接口的显示器	1	
	HDMI 视频线	2	
软件	Vivado Design Suite	1	版本: 2014.4
	视频接口 IP 库	1	FPGA-Image-Library.zip*

*注: FPGA-Image-Library 为戴天宇开发的一个开源图像处理 IP 库, 该 IP 库遵循 LGPL, 详情请见: <http://fil.dtysky.moe>

1.4 实验原理

该实验的连接方式如下图所示:

	标题	文档编号	版本	页
	Lab7: 图像处理滤波器实验 4	XDI-SWORD-IMG-007	1.0	7 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/5/13		

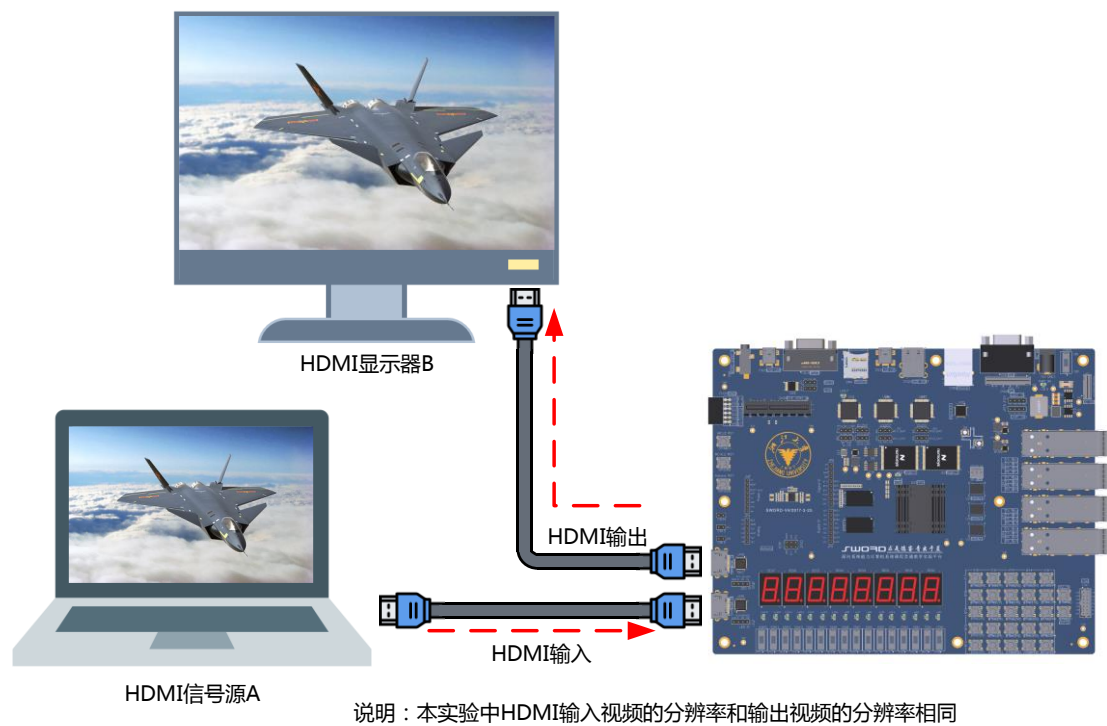


图 1-1 实验连接示意图

实验利用了 1 个 IP 来实现对比度变换：ContrastTransform。其中：对比度变换实验 IP 连接示意图如下图所示：

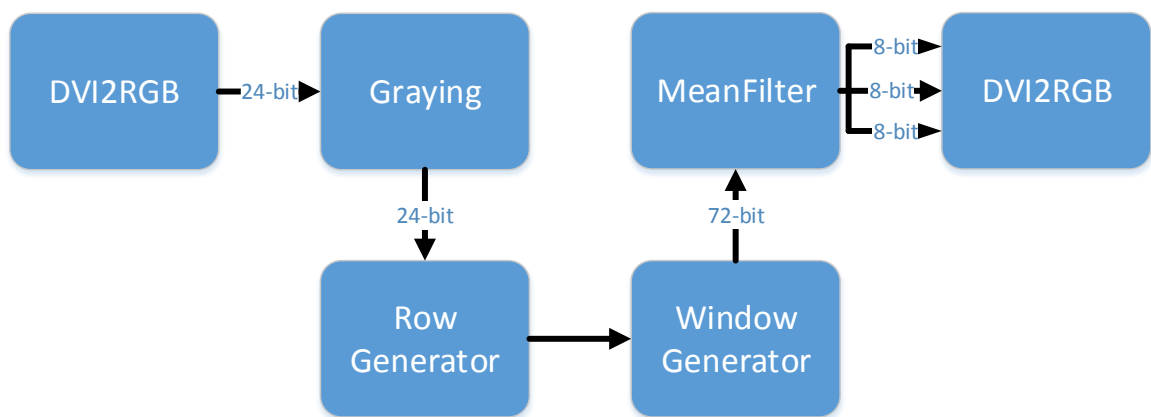


图 1-2 图像对比度调节连接示意图

RowGenerator 这个 IP 的作用是行缓存。

	标题	文档编号	版本	页
	Lab7: 图像处理滤波器实验 4	XDI-SWORD-IMG-007	1.0	8 of 31
	作者	修改日期	公开	
Joseph Xu		2018/5/13		

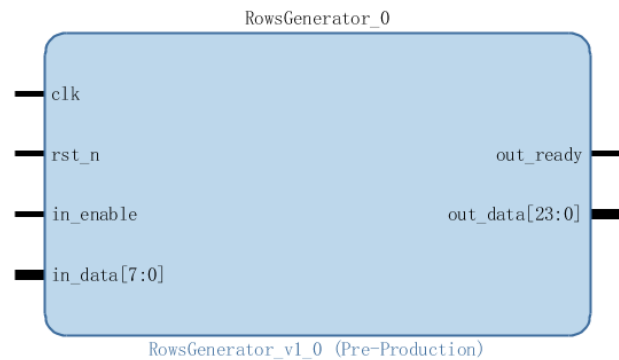


图 1-3 RowGenerator IP

该 IP 的端口信号定义如下表所示：

表 1 RowGenerator IP 端口列表

信号名	方向	宽度	含义
clk	I	1	Clock.
rst_n	I	1	Reset, active low.
in_enable	I	1	Input data enable, it works as fifo0's wr_en.
in_data	I	Color_Width	Input data, it must be synchronous with in_enable.
out_ready	O	1	Output data ready, in both two mode, it will be high while the out_data can be read.
out_data	O	rows_width * color_width	Output data, it will be synchronous with out_ready.The lowest color_width-bits of this are the first row!

而 WindowGenerator 这个 IP 的作用是滑动窗口缓存。

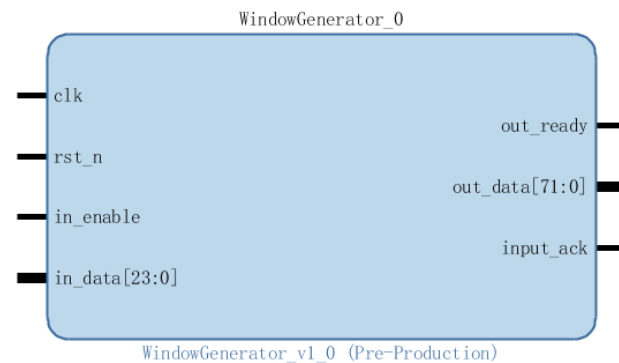


图 1-4 WindowGenerator IP

该 IP 的端口信号定义如下表所示：

	标题	文档编号	版本	页
	Lab7: 图像处理滤波器实验 4	XDI-SWORD-IMG-007	1.0	9 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/5/13		

表 2 WindowGenerator IP 端口列表

信号名	方向	宽度	含义
clk	I	1	Clock.
rst_n	I	1	Reset, active low.
in_enable	O	1	Input data enable, in pipeline mode, it works as another rst_n, in req-ack mode, only it is high will in_data can be really changes.
in_data	O	color_width * window_width	Input data, it must be synchronous with in_enable.
out_ready	O	1	Output data ready, in both two mode, it will be high while the out_data can be read.
out_data	O	color_width* window_width *window_width	Output data, it will be synchronous with out_ready.
input_ack	O	1	Input ack, only used for req-ack mode, this port will give a ack while the input_data received.

而 MeanFilter 这个 IP 的作用是（算术）均值滤波器。

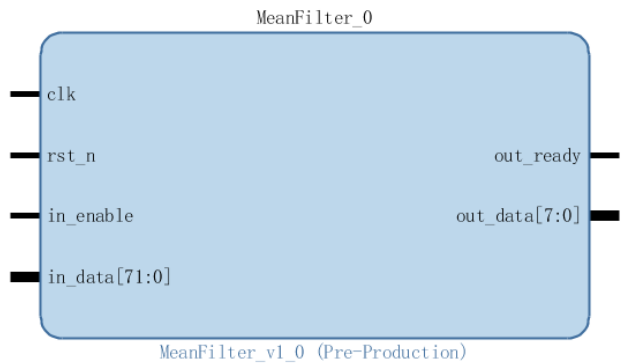



图 1-5 MeanFilter IP

该 IP 的端口信号定义如下表所示：

表 3 MeanFilter IP 端口列表

信号名	方向	宽度	含义
clk	I	1	Clock.
rst_n	I	1	Reset, active low.
in_enable	I	1	Input data enable, in pipeline mode, it works as another rst_n, in req-ack

信号名	方向	宽度	含义
			mode, only it is high will in_data can be really changes.
in_data	I	color_width * window_width * window_width	Input data, it must be synchronous with in_enable.
out_ready	O	1	Output data ready, in both two mode, it will be high while the out_data can be read.
out_data	O	color_width	Output data, it will be synchronous with out_ready.

	标题	文档编号	版本	页
	Lab7: 图像处理滤波器实验 4	XDI-SWORD-IMG-007	1.0	11 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/5/13		

2. 二值图腐蚀实验流程

本章将详细描述如何在 Vivado 2014.4 的环境下完成实验。请耐心等待，仔细按照图示和文字说明进行操作。

2.1 操作步骤

- 1 由于本实验是在实验 6 的基础上进行扩展，所以我们将之前的实验部分复制 1 份，具体做法为在 D:\ImageLabs 文件夹下，将鼠标左键选中 lab6，然后按住 Ctrl 键不放，并拖拽到空白处，这样得到一个 lab6 的副本，如下图所示：

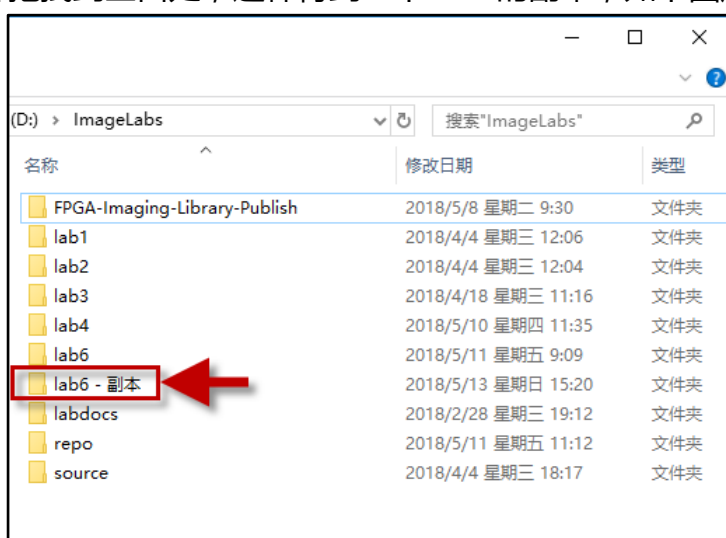


图 2-1 复制一个实验 6 副本

然后将 lab4 的副本重命名为 lab6，如下图所示，至此我们就可以在 lab6 文件夹里开始我们的实验内容：

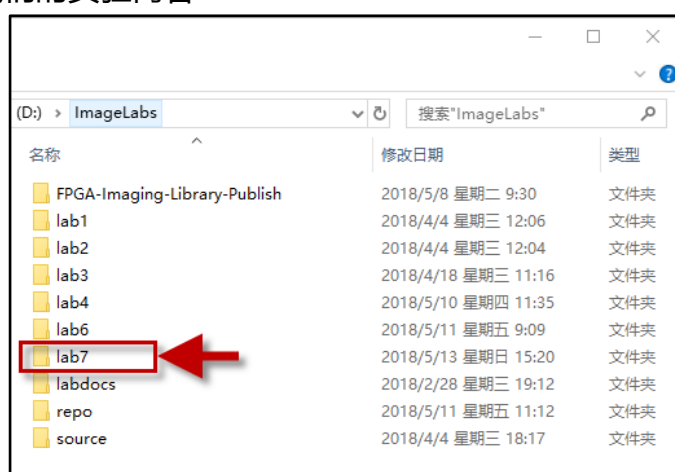



图 2-2 重命名实验目录

- 2 接着启动 Vivado 2014.4，在启动界面选择 Open Project，如下图所示：

	标题	文档编号	版本	页
	Lab7: 图像处理滤波器实验 4	XDI-SWORD-IMG-007	1.0	12 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/5/13		

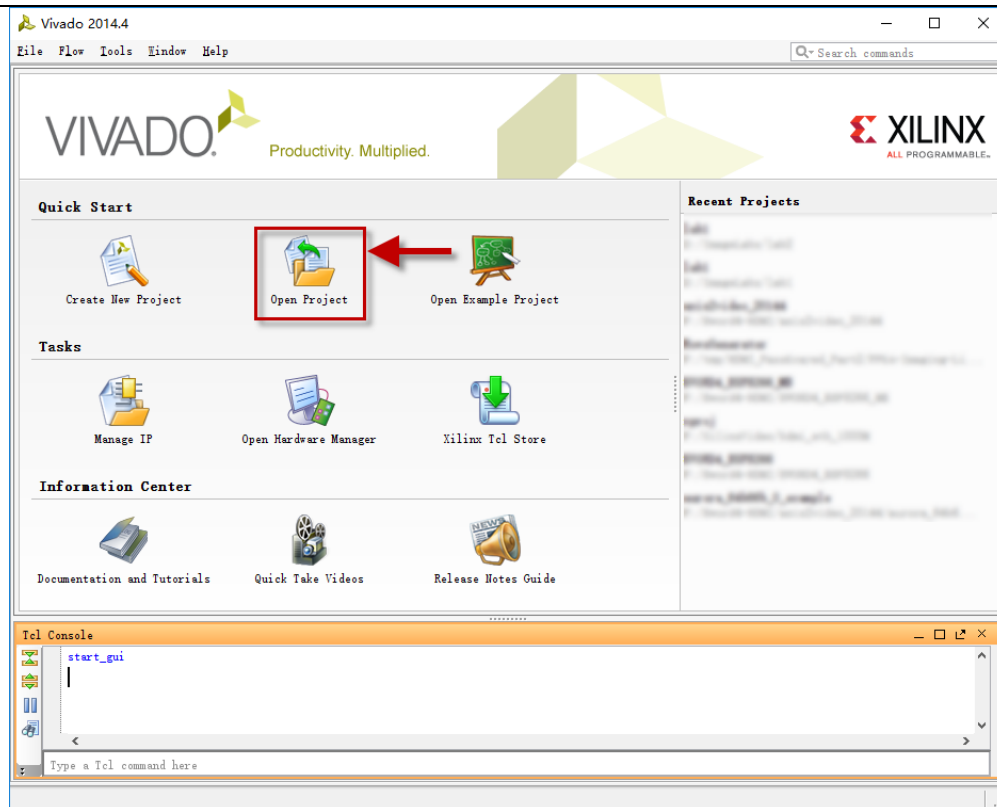


图 2-3 启动 Vivado

- 3 然后在选择对话框中 找到之前的 lab7 (即 D:\ImageLabs\lab7) 然后选择 lab1.xpr 文件, 点击 OK, 打开工程, 整个过程如下图所示:

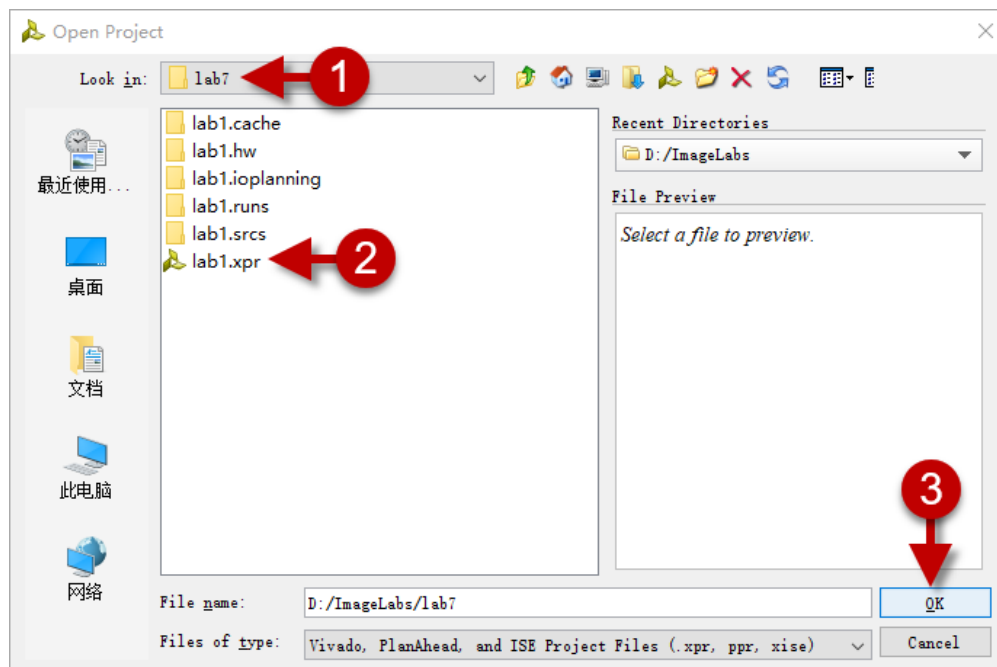



图 2-4 打开工程

- 4 然后在 Vivado 的主界面, 点击 Open Block Design, 这时会在主界面右边区域看到之前实验 6 的 IP 结构, 如下图所示:

	标题	文档编号	版本	页
	Lab7: 图像处理滤波器实验 4	XDI-SWORD-IMG-007	1.0	13 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/5/13		

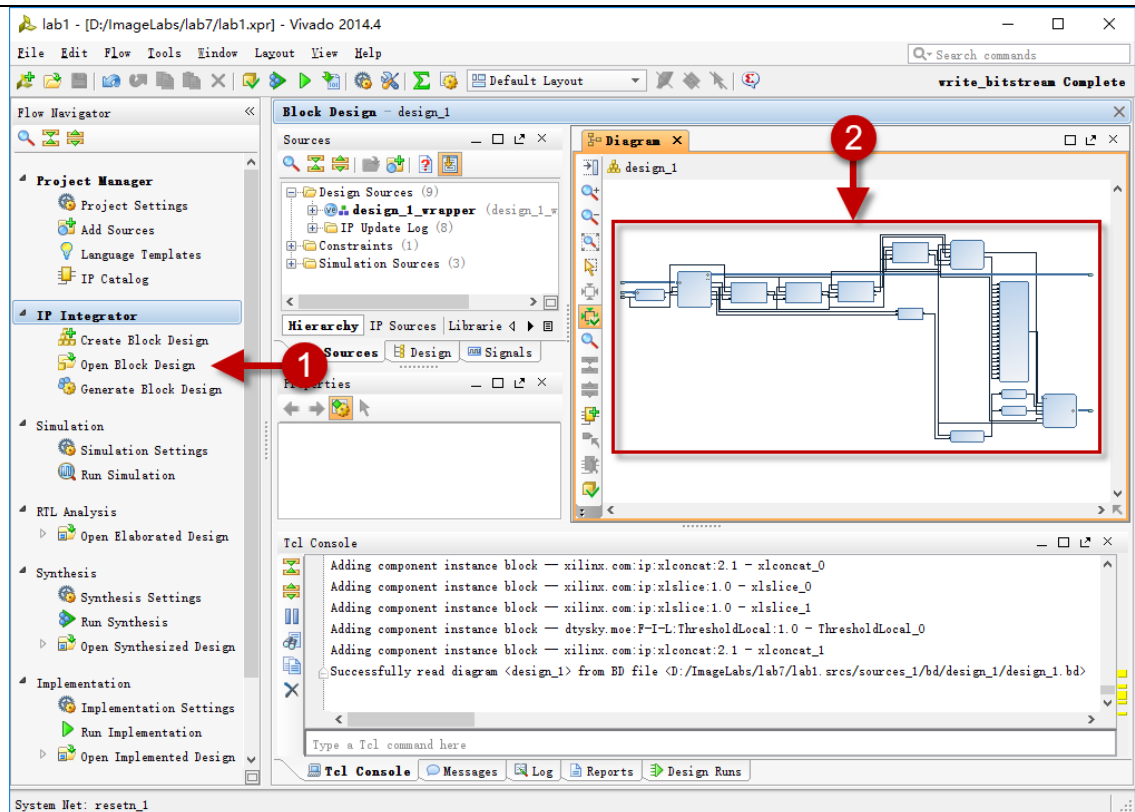


图 2-5 实验初始视图

- 5 在此基础上，我们开始添加 IP，点击左边栏的 Add IP 图标，然后在弹出的搜索框中，输入 ero，这时能看到搜索结果中有个 ErosionDilationBin 的 IP，双击它进行添加，整个过程如下图所示：

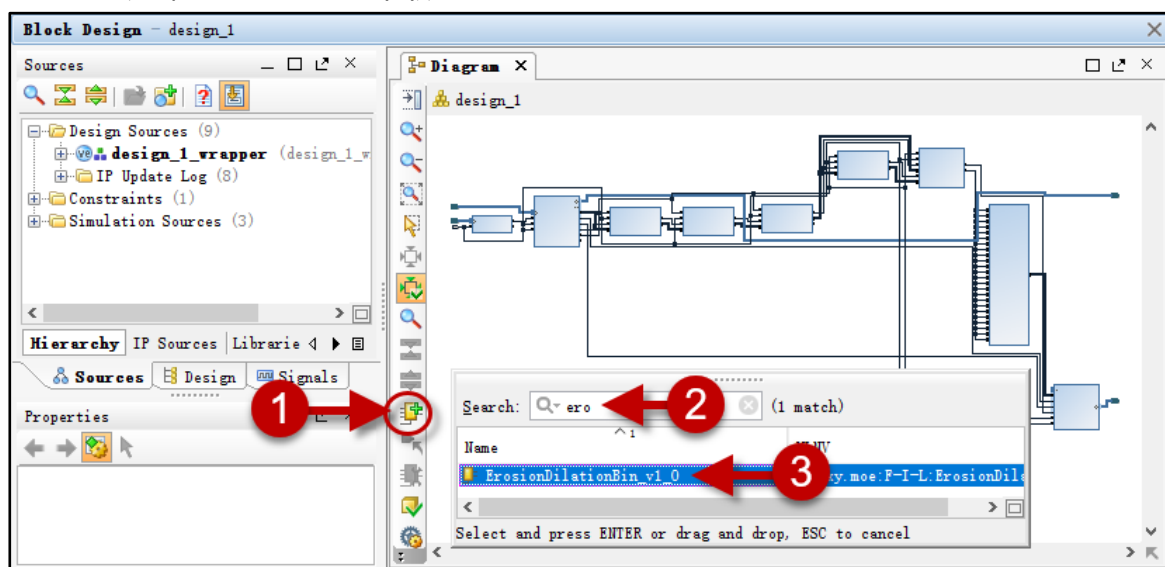



图 2-6 添加局部阈值化 IP

- 6 添加后的 IP 视图如下图所示：

	标题	文档编号	版本	页
	Lab7: 图像处理滤波器实验 4	XDI-SWORD-IMG-007	1.0	14 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/5/13		

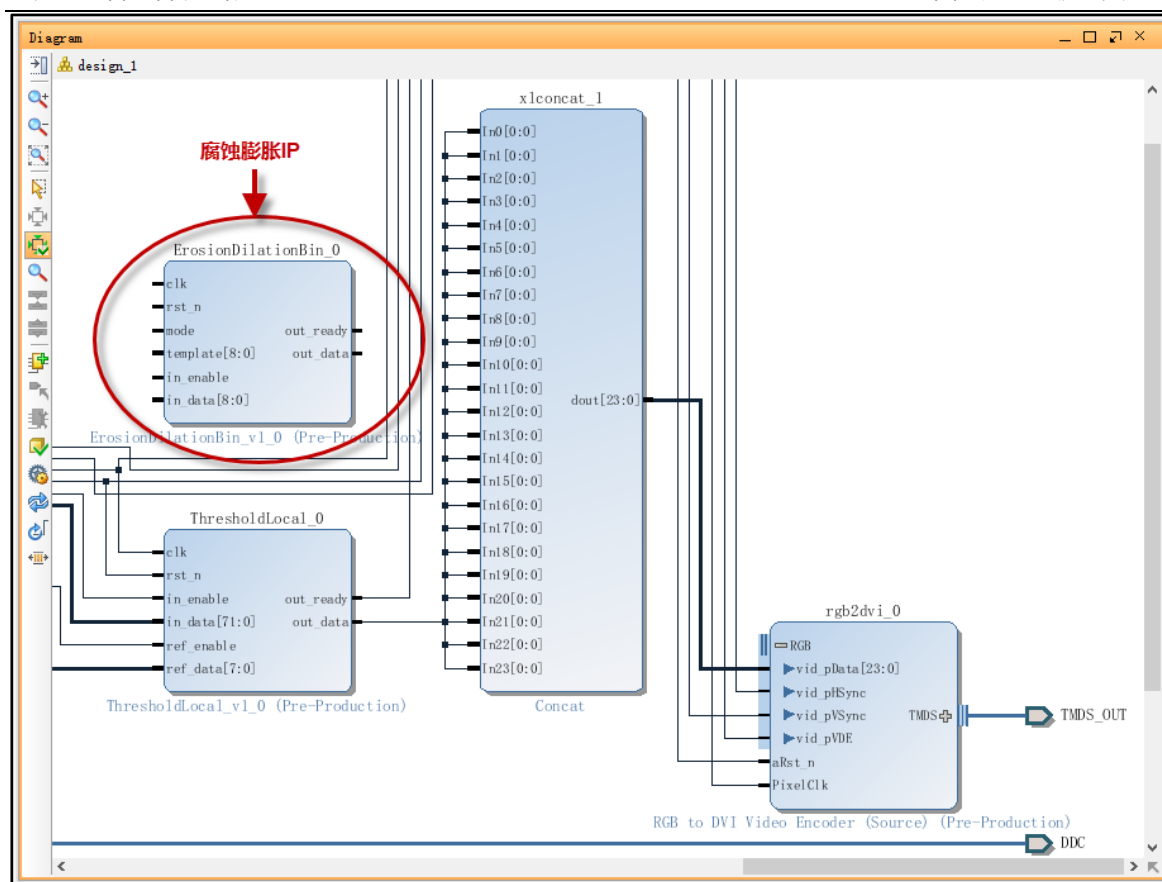


图 2-7 添加腐蚀膨胀 IP 后的视图

7 双击 ErosionDilationBin_0 这个 IP 进行配置，配置参数如下图所示：

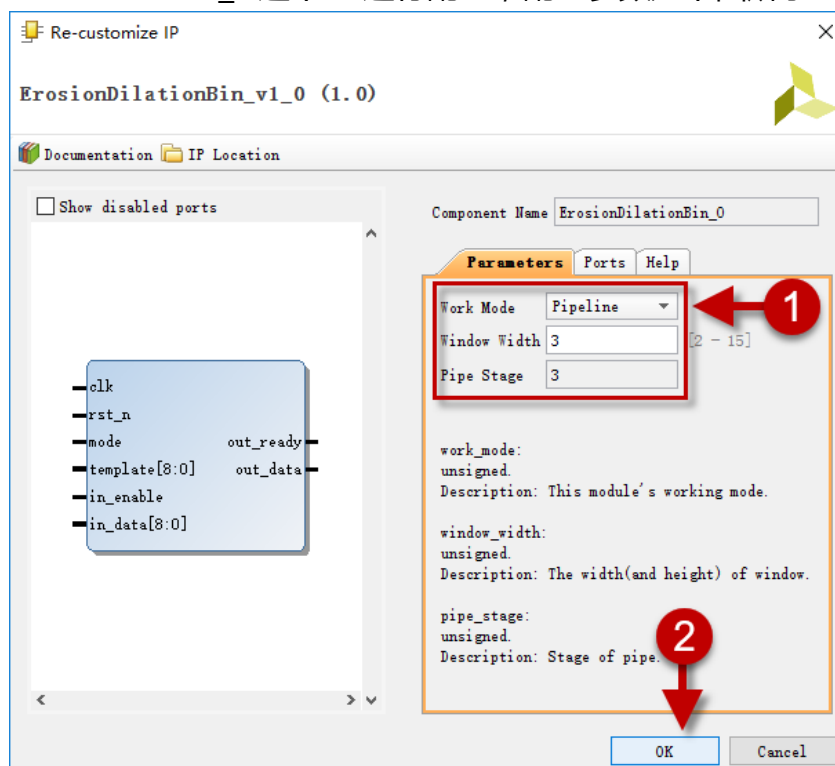


图 2-8 腐蚀膨胀 IP 配置

	标题	文档编号	版本	页
	Lab7: 图像处理滤波器实验 4	XDI-SWORD-IMG-007	1.0	15 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/5/13		

- 8 由于腐蚀膨胀 IP 是对二值图像进行操作，所以 3x3 窗口下的数据宽度为 9 位，即 in_data[8:0]，由于之前的滑动窗口数据位宽为 72 位的，为此我们需要在局部阈值化的输出后再接一个 3x3 的滑动窗口，该滑动窗口的输出作为腐蚀膨胀 IP 的输入。

点击左边栏的 Add IP 图标，然后在弹出的搜索框中，输入 rows，这时能看到搜索结果中有个 RowsGenerator 的 IP，双击它进行添加，整个过程如下图所示：

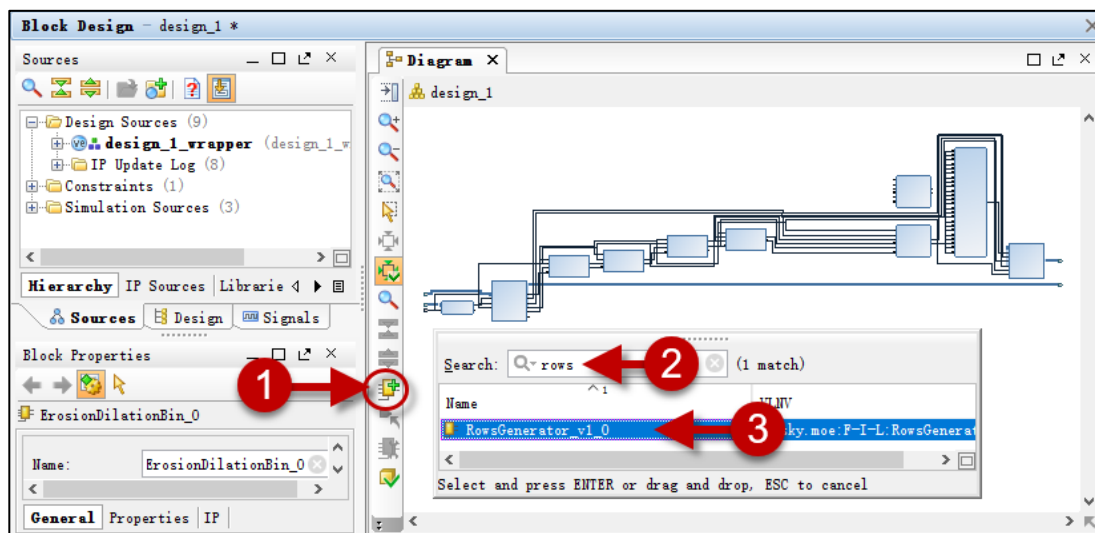


图 2-9 添加 RowsGenerator IP

接着继续添加 IP，点击左边栏的 Add IP 图标，在弹出的搜索框中，输入 Window，这时能看到搜索结果中有个 WindowGenerator 的 IP，双击它进行添加，整个过程如下图所示：

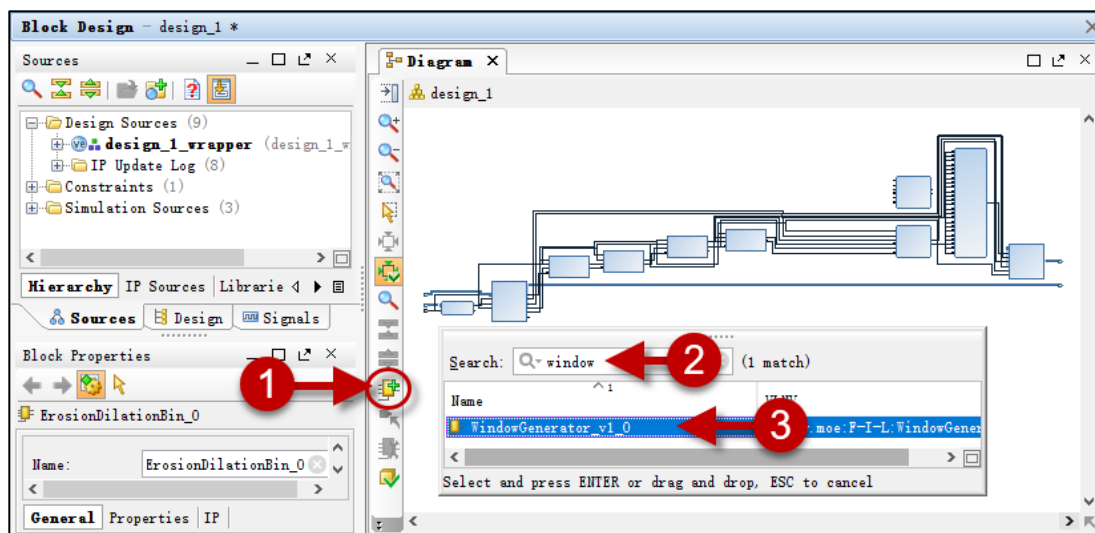



图 2-10 添加 WindowGenerator IP

- 9 接着我们依次对这 2 个 IP 进行配置。

首先配置 RowsGenerator IP，双击 RowsGenerator_1 这个 IP，然后按照如下方式

	标题	文档编号	版本	页
	Lab7: 图像处理滤波器实验 4	XDI-SWORD-IMG-007	1.0	16 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/5/13		

进行配置：

- Rows Width : 3
- Im Width : 1274
- **Color Width : 1**
- Im Width Bits : 11

确认上述设置后，点击 OK 确定，整个过程如下图所示：

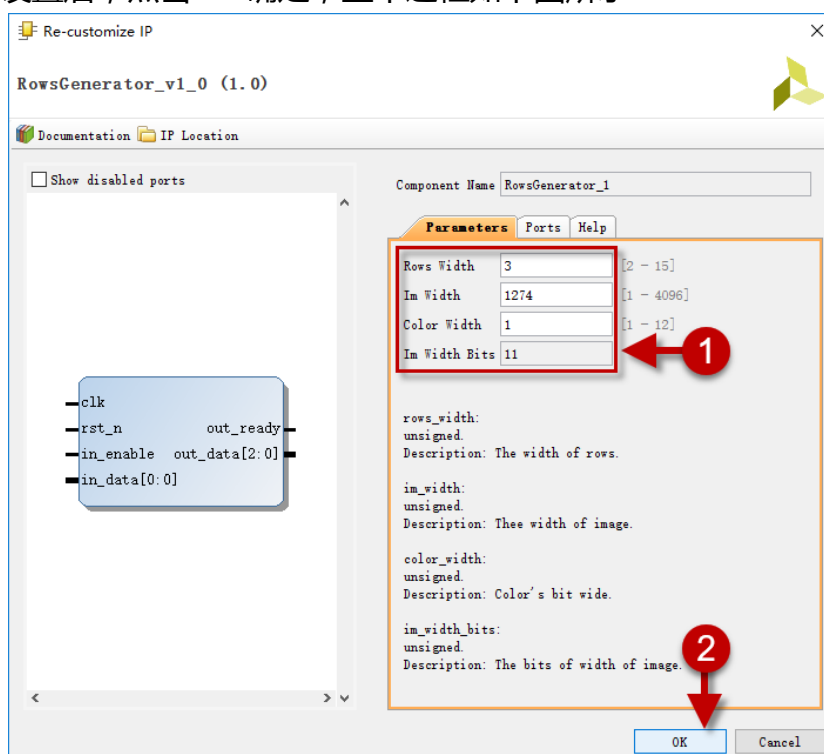



图 2-11 RowsGenerator IP 配置

接着配置 WindowGenerator IP，双击 WindowGenerator_1 这个 IP，然后按照如下方式进行配置：

- Work Mode : Pipeline
- Window Width : 3
- **Color Width : 1**
- Window Width Half : 2

确认上述设置后，点击 OK 确定，整个过程如下图所示：

	标题	文档编号	版本	页
	Lab7: 图像处理滤波器实验 4	XDI-SWORD-IMG-007	1.0	17 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/5/13		

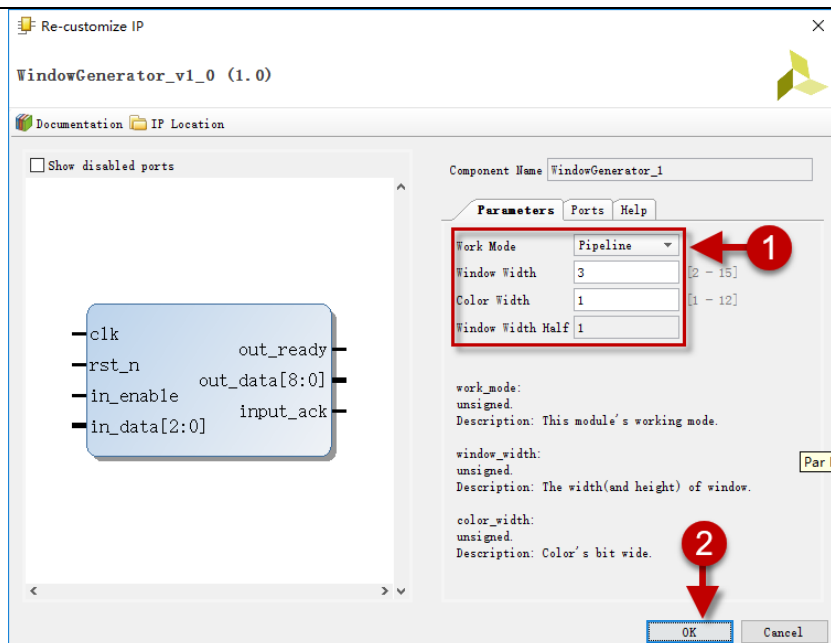


图 2-12 WindowGenerator IP 配置

10 同时为了突出腐蚀膨胀的效果,我们将局部二值化的输出进行一下反色处理,这样主背景就为黑色,线条为白色。为此我们需要在局部阈值化的输出后再接一个反色 IP,该 IP 的输出作为刚才添加的滑动窗口的输入。

点击左边栏的 Add IP 图标,然后在弹出的搜索框中,输入 rev,这时能看到搜索结果中有个 ColorReversal 的 IP,双击它进行添加,整个过程如下图所示:

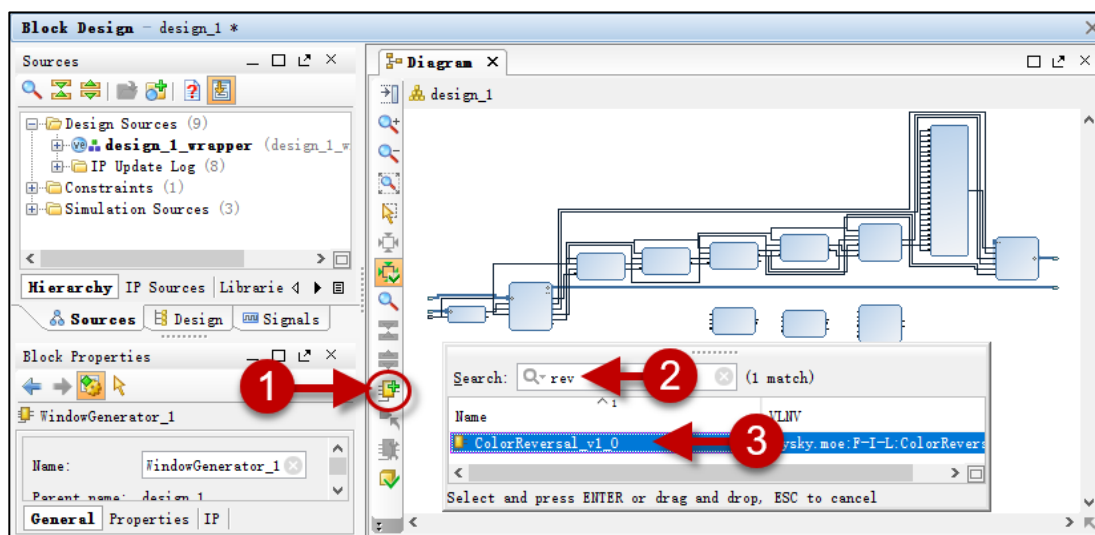


图 2-13 添加反色 IP

接着双击这个 IP 进行配置,按照如下方式进行配置:

- Work Mode : Pipeline
- **Color Channels : 1**
- **Color Width : 1**

确认上述设置后,点击 OK 确定,整个过程如下图所示:

	标题	文档编号	版本	页
	Lab7: 图像处理滤波器实验 4	XDI-SWORD-IMG-007	1.0	18 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/5/13		

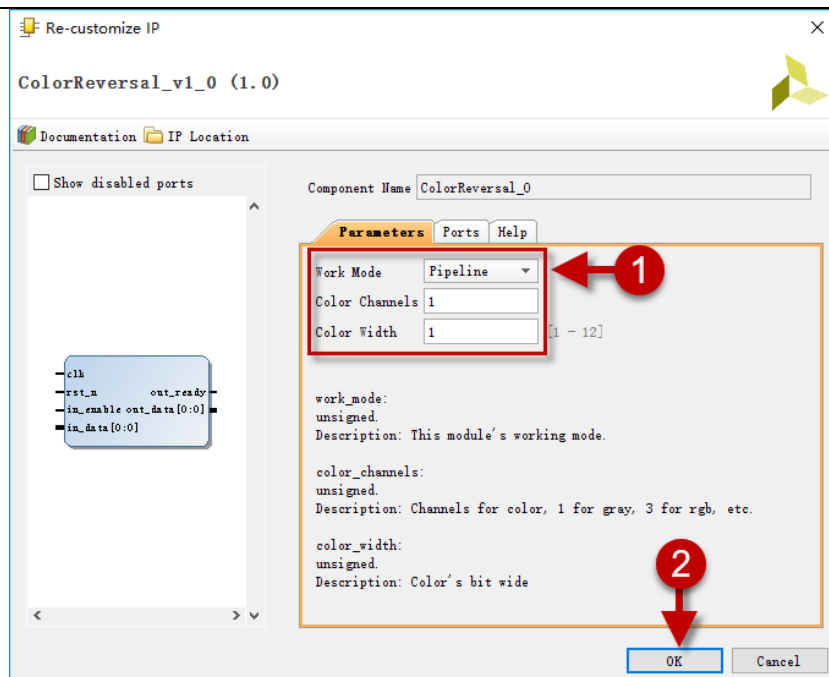


图 2-14 配置反色 IP

- 11 然后我们开始连接这些 IP。我们需要先将之前的数据流先断开,为此我们先用鼠标左键选中 ThresholdLocal IP 的 out_data 端口,此时会看到该信号高亮为浅黄色 (注意一定不要选中整个 IP),然后鼠标右键单击,在弹出菜单中选择 Disconnect Pin, 整个过程如下图所示:

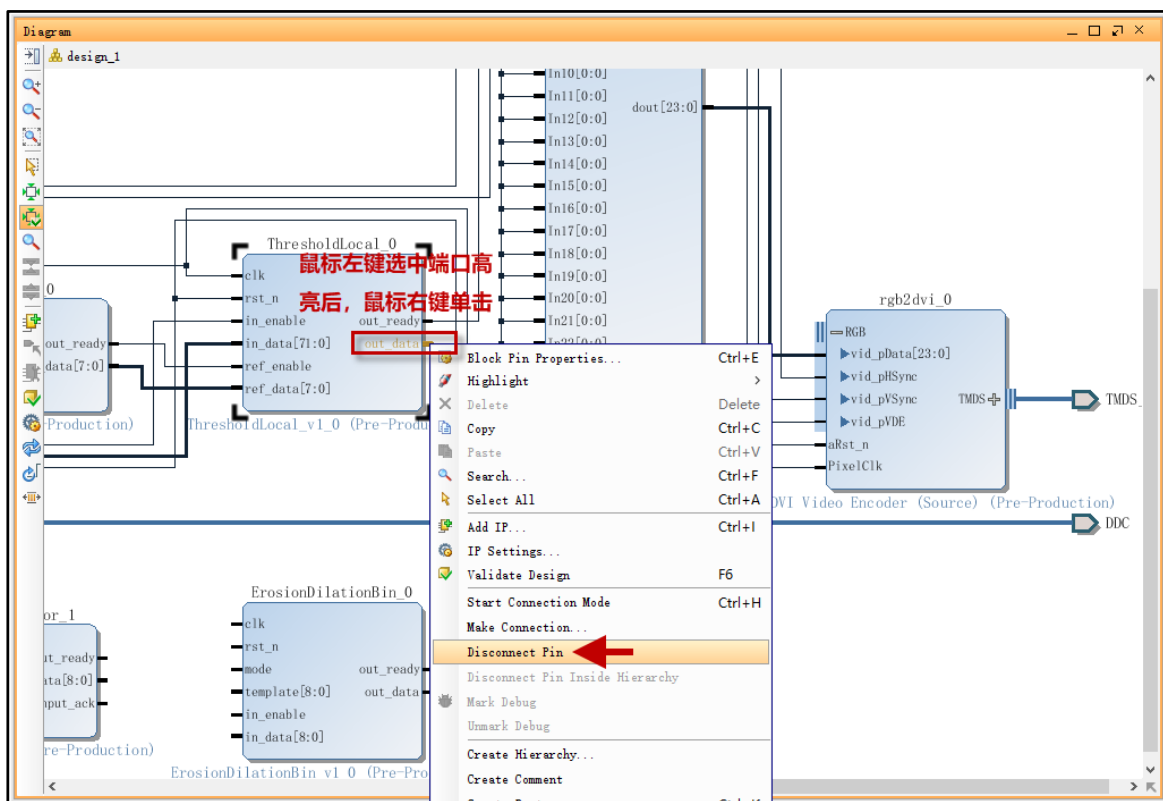



图 2-15 断开 ThresholdLocal IP 的 out_data 端口

	标题	文档编号	版本	页
	Lab7: 图像处理滤波器实验 4	XDI-SWORD-IMG-007	1.0	19 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/5/13		

然后按照同样的方法断开 ThresholdLocal IP 的 out_ready 端口，断开后的 IP 视图如下图所示：

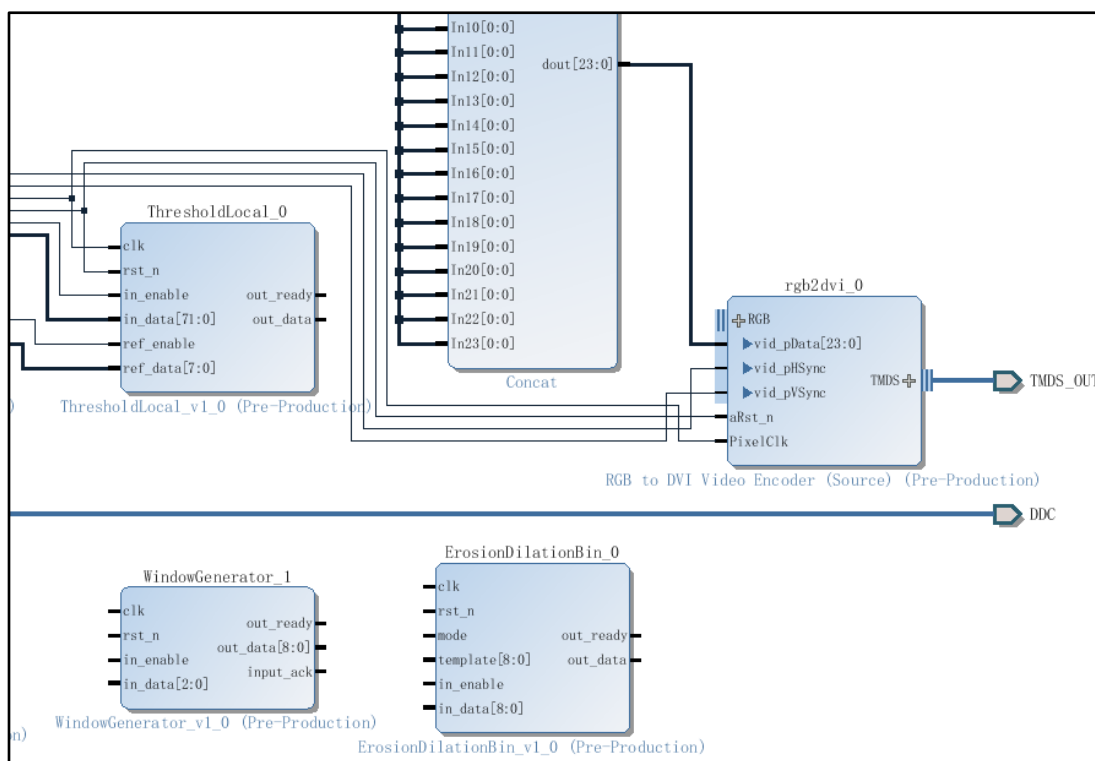


图 2-16 断开端口后视图

接着我们将按照如下方式先对 ColorReversal IP 进行连接：

ColorReversal_0:clk → dvi2rgb_0:PixelClk

ColorReversal_0:rst_n → clk_wiz_0:resetn

ColorReversal_0:in_enable → ThresholdLocal_0:out_ready


ColorReversal_0:in_data[0:0] → ThresholdLocal_0:out_data

ColorReversal_0:out_ready → RowsGenerator_1:in_enable

ColorReversal_0:out_data[0:0] → RowsGenerator_1:in_data[0:0]

连接后的效果如下图所示，请仔细检查各 IP 的端口连接是否正确，为了方便核对，下图各种连接的高亮色图以示区别：

提示：下图仅作为检查连接使用，读者完全不必也按照图示颜色进行标注！！（下同）

	标题	文档编号	版本	页
	Lab7: 图像处理滤波器实验 4	XDI-SWORD-IMG-007	1.0	20 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/5/13		

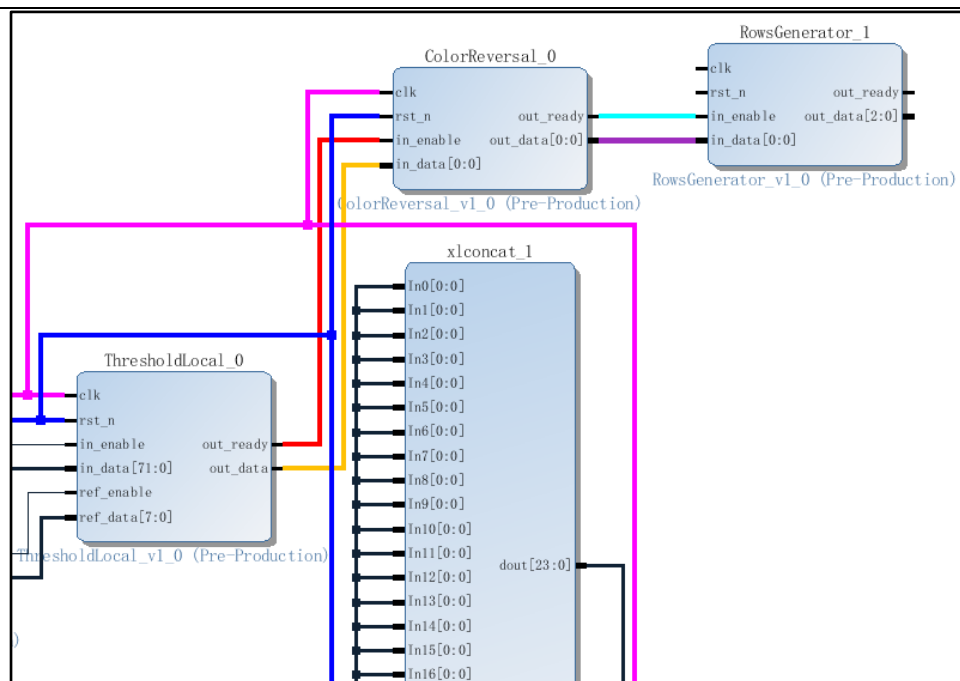


图 2-17 连接 ColorReversal 和 RowsGenerator

12 接着连接 RowsGenerator_1 和 WindowGenerator_1 :

RowsGenerator_1:clk → dvi2rgb_0:PixelClk

RowsGenerator_1:rst_n → clk_wiz_0:resetn

RowsGenerator_1:out_ready → WindowGenerator_1:in_enable

RowsGenerator_1:out_data[2:0] → WindowGenerator_1:in_data[2:0]

WindowGenerator_1:clk → dvi2rgb_0:PixelClk

WindowGenerator_1:rst_n → clk_wiz_0:resetn

连接后的效果如下图所示，请仔细检查各 IP 的端口连接是否正确：

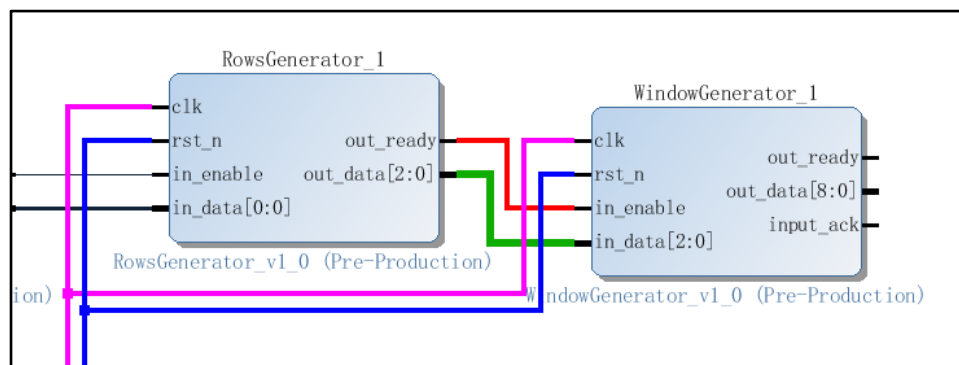


图 2-18 连接 RowsGenerator 和 WindowGenerator

13 接着连接 WindowGenerator_1 和 ErosionDilationBin_0 :

ErosionDilationBin_0:clk → dvi2rgb_0:PixelClk

ErosionDilationBin_0:rst_n → clk_wiz_0:resetn


ErosionDilationBin_0:in_enable → WindowGenerator_1: out_ready

ErosionDilationBin_0:in_data[8:0] → WindowGenerator_1: out_data [8:0]

ErosionDilationBin_0:out_ready → rgb2dvi_0:vid_pVDE

ErosionDilationBin_0:out_data → xlconcat_0:In0[0:0]

连接后的效果如下图所示，请仔细检查各 IP 的端口连接是否正确：

	标题	文档编号	版本	页
	Lab7: 图像处理滤波器实验 4	XDI-SWORD-IMG-007	1.0	21 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/5/13		

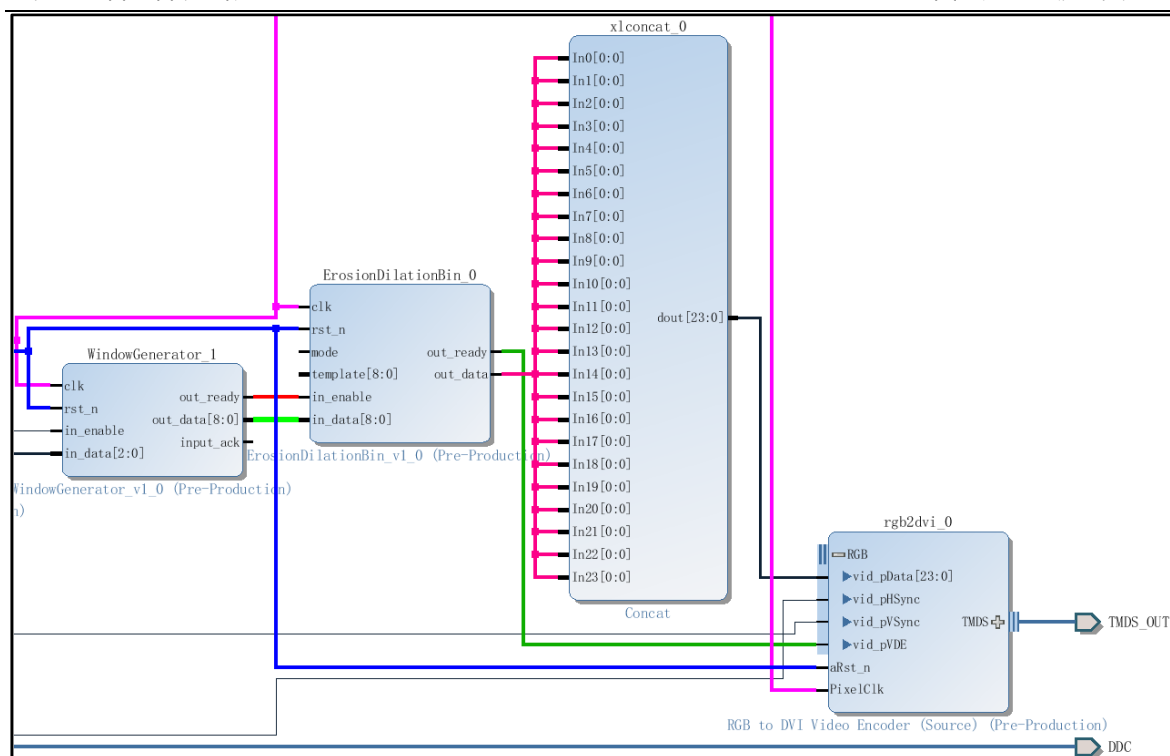


图 2-19 连接腐蚀膨胀 IP

最后我们需要对腐蚀膨胀 IP 进行模式和模板配置，为此我们需要添加 2 个常数 IP 作为腐蚀膨胀 IP 的输入。点击左边栏的 Add IP 图标，然后在弹出的搜索框中，输入 constant，这时能看到搜索结果中有个 constant 的 IP，双击它进行添加，整个过程如下图所示：

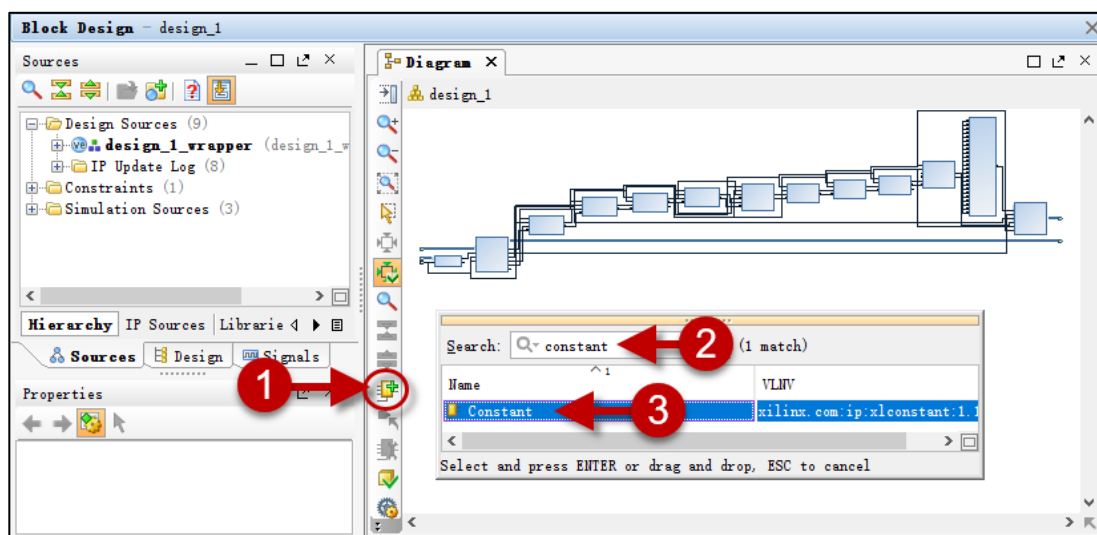



图 2-20 添加常数 IP

注意：一共需要添加 2 个常数 IP：xlconstant_0 和 xlconstant_1，添加后的效果如下图所示：

	标题	文档编号	版本	页
	Lab7: 图像处理滤波器实验 4	XDI-SWORD-IMG-007	1.0	22 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/5/13		

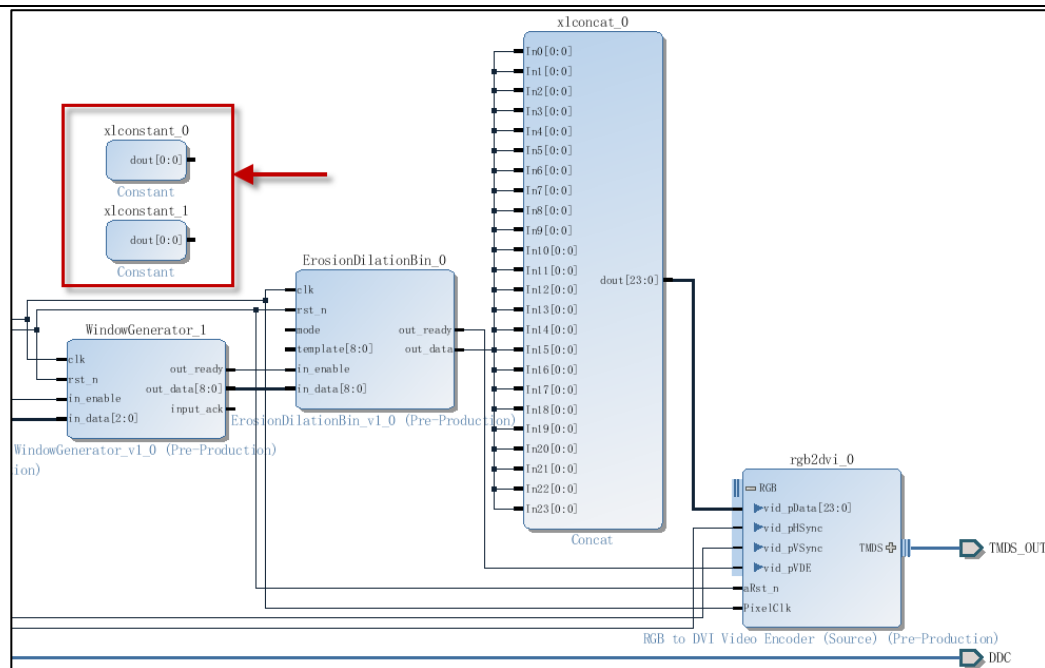


图 2-21 添加 2 个常数 IP

然后分别配置这两个常数 IP。首先双击配置 xlconstant_0，将其作为腐蚀模式输入，按照如下方式进行配置：

- Const Width : 1
- Const Val : 1

确认上述设置后，点击 OK 确定，整个过程如下图所示：

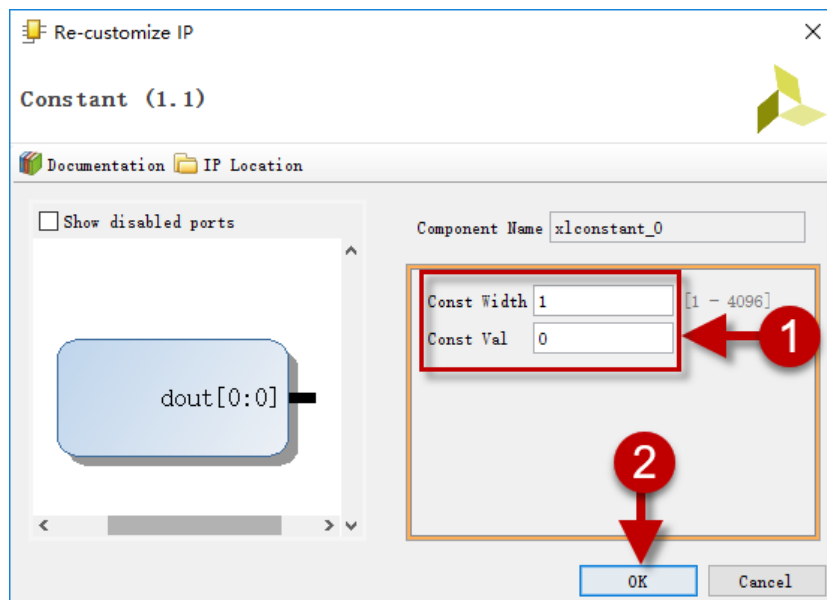


图 2-22 配置腐蚀模式

接着双击配置 xlconstant_1，将其作为腐蚀模板输入，按照如下方式进行配置：

- Const Width : 9
- Const Val : 432 (十进制 432=二进制 110110000)

XINGDENG	标题	文档编号	版本	页
	Lab7: 图像处理滤波器实验 4	XDI-SWORD-IMG-007	1.0	23 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/5/13		

确认上述设置后，点击 OK 确定，整个过程如下图所示：

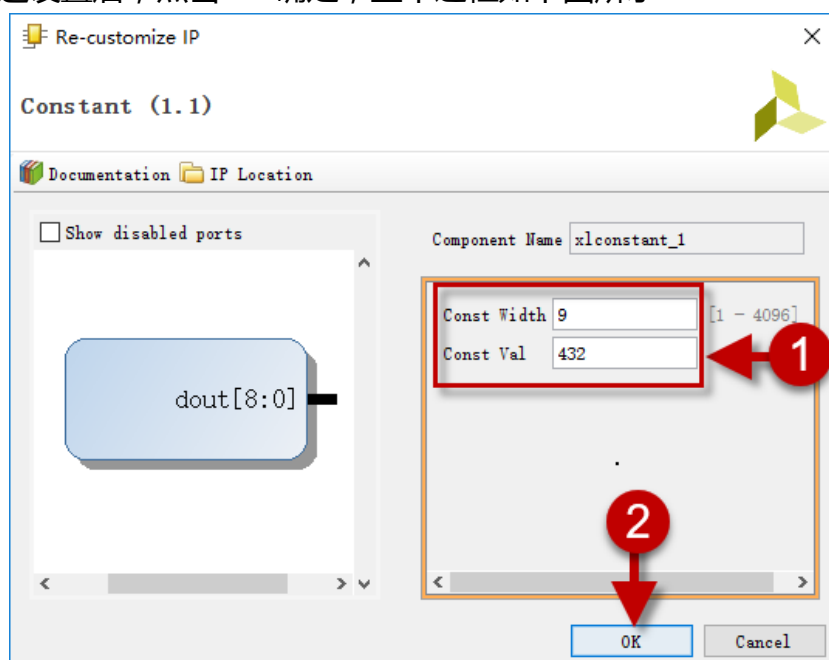


图 2-23 配置腐蚀模式

完成配置后进行 IP 的连接，连接后的效果如下图所示：

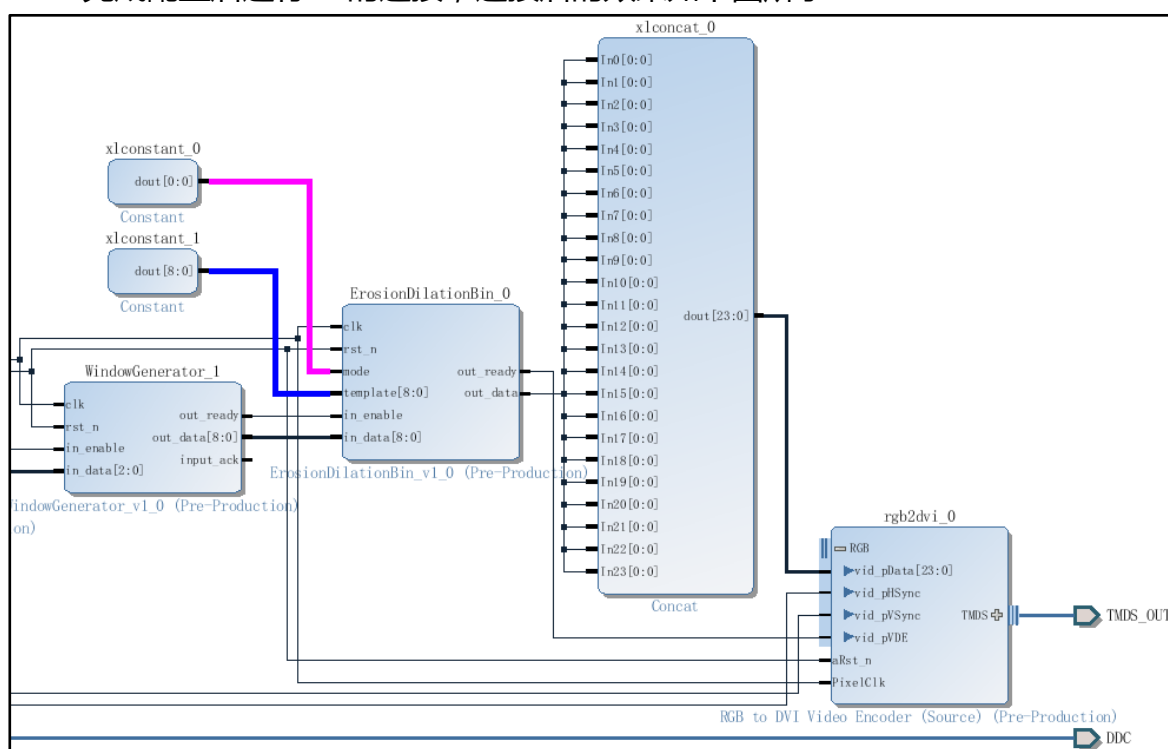



图 2-24 端口连接检查

- 14 连接检查无误后，即可保存 IP 模块化设计，在 Vivado 主界面点击保存图标，如下图所示：

	标题	文档编号	版本	页
	Lab7: 图像处理滤波器实验 4	XDI-SWORD-IMG-007	1.0	24 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/5/13		

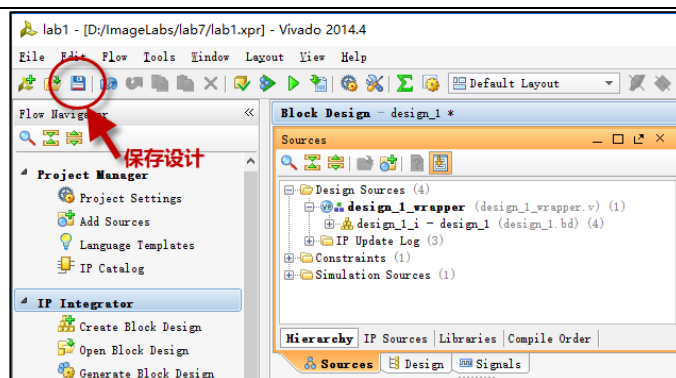


图 2-25 保存设计

接着在 Source 子窗口中展开 design_1_wrapper，选中 design_1.bd，鼠标右键单击，在弹出的菜单中选择 Create HDL Wrapper，整个过程如下图所示：

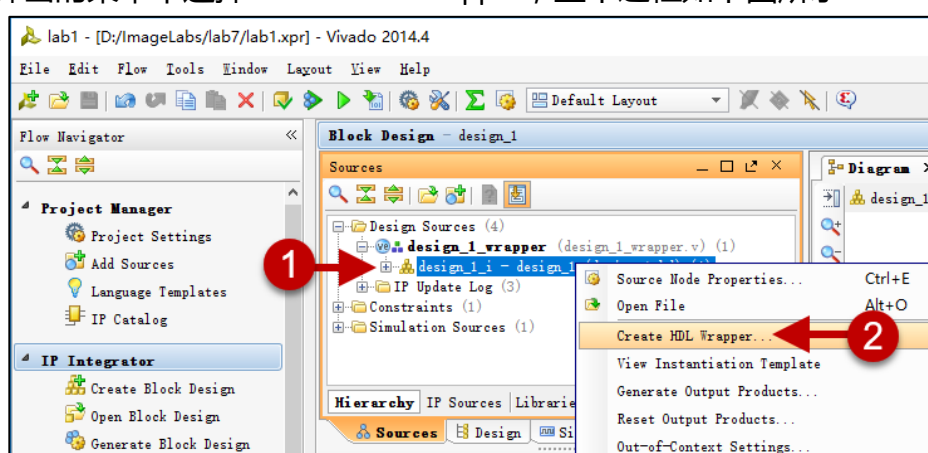


图 2-26 创建实验顶层 Wrapper 文件

接着在弹出的对话框中，保持默认的选项不变，即选择 Let Vivado manage wrapper and auto-update，然后点击 OK，如下图所示：

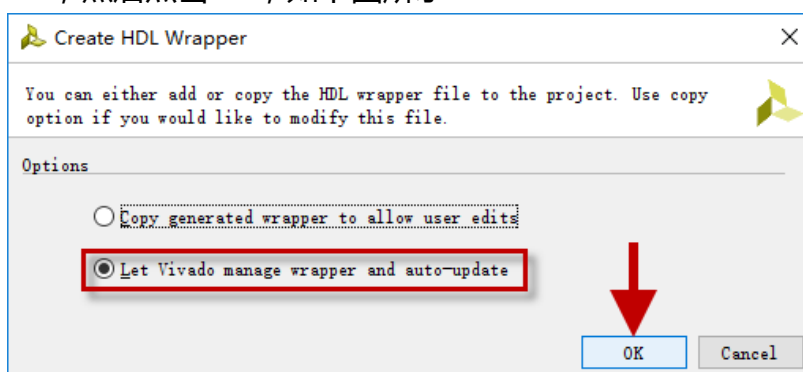



图 2-27 自动更新顶层文件

接着我们还要对 lab1.xdc 文件进行修改。为此，先在 Vivado 的主界面选中 Project Manager，接着在右边的 Source 窗口栏中依次展开 Constraints → constrs_1，在展开后的最下层可以看到 lab1.xdc 文件，双击这个文件打开，整个过程如下图所示：

	标题	文档编号	版本	页
	Lab7: 图像处理滤波器实验 4	XDI-SWORD-IMG-007	1.0	25 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/5/13		

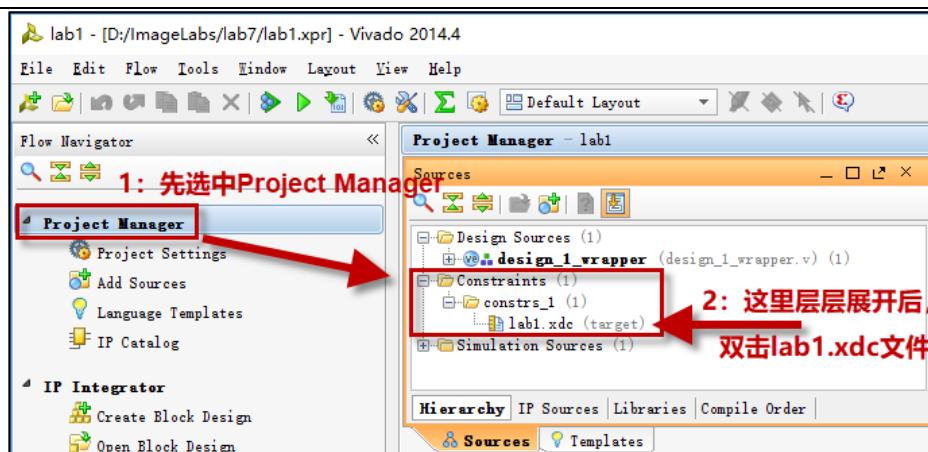


图 2-28 打开 lab1.xdc 文件

在该文件的第一行，加上如下 4 行语句：

```
set_property DONT_TOUCH true [get_cells design_1_i/xlconstant_0]
set_property DONT_TOUCH true [get_cells design_1_i/xlconstant_0/inst]
set_property DONT_TOUCH true [get_cells design_1_i/xlconstant_1]
set_property DONT_TOUCH true [get_cells design_1_i/xlconstant_1/inst]
```

添加后点击窗口左边的保存图标，保存修改。整个过程如下图所示：

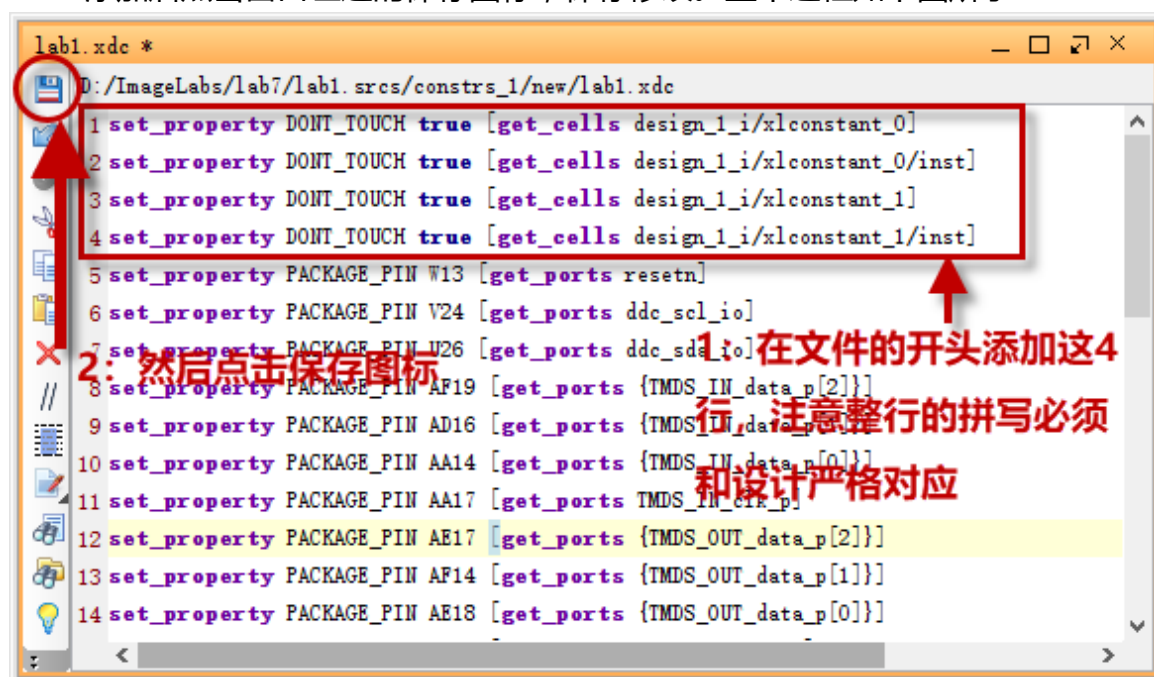


图 2-29 修改 lab1.xdc 文件

之后在 Vivado 主界面点击 Generate Bitstream，生成 bit 文件，如下图所示：

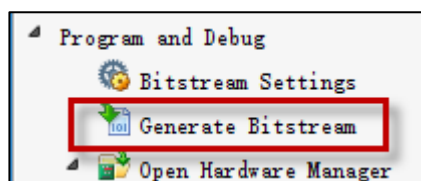


图 2-30 Generate Bitstream

在弹出的提示框中直接点 Yes 确认并继续，如下图所示：

	标题	文档编号	版本	页
	Lab7: 图像处理滤波器实验 4	XDI-SWORD-IMG-007	1.0	26 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/5/13		

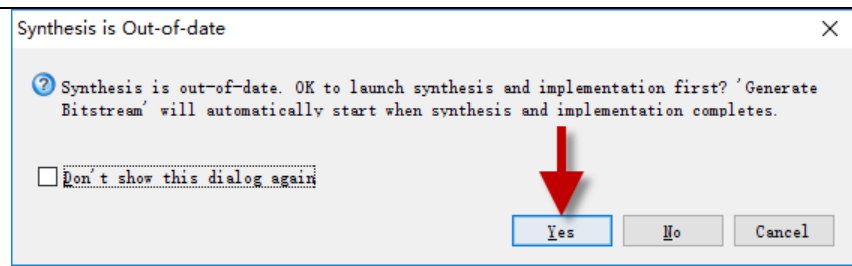


图 2-31 点击 Yes 确认生成 bit 文件

大约经过 10 分钟后，Vivado 会弹出 Bitstream Generation Completed 的提示框，表示 bit 文件完成，选择 Open Hardware Manager，然后点击 OK，如下图所示：

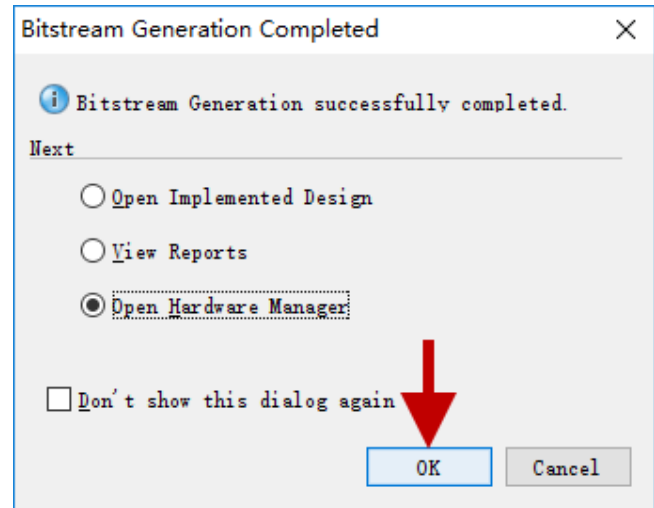


图 2-32 打开 Hardware Manager

接着我们需要对 SWORD4.0 硬件平台进行连接，根据下图示意依次进行如下操作：

- 1) 将电源线接上 SWORD4.0，注意此时 SWORD4.0 的开关不要打开；
- 2) 将下载器模块插到 SWORD4.0 的 CN7-JTAG 处，并将下载器的 USB 端口连到电脑；
- 3) 用一根 HDMI 线将 SWORD4.0 和 HDMI 信号源连接上；
- 4) 用一根 HDMI 线将 SWORD4.0 和 HDMI 显示器连接上；
- 5) 打开电源开关
- | | | | | |
|--|-------------------|-------------------|-----|----------|
| | 标题 | 文档编号 | 版本 | 页 |
| | Lab7: 图像处理滤波器实验 4 | XDI-SWORD-IMG-007 | 1.0 | 27 of 31 |
| | 作者 | 修改日期 | 公开 | |
| | Joseph Xu | 2018/5/13 | | |
- Copyright © 2018 XingDeng, Inc. All rights reserved.

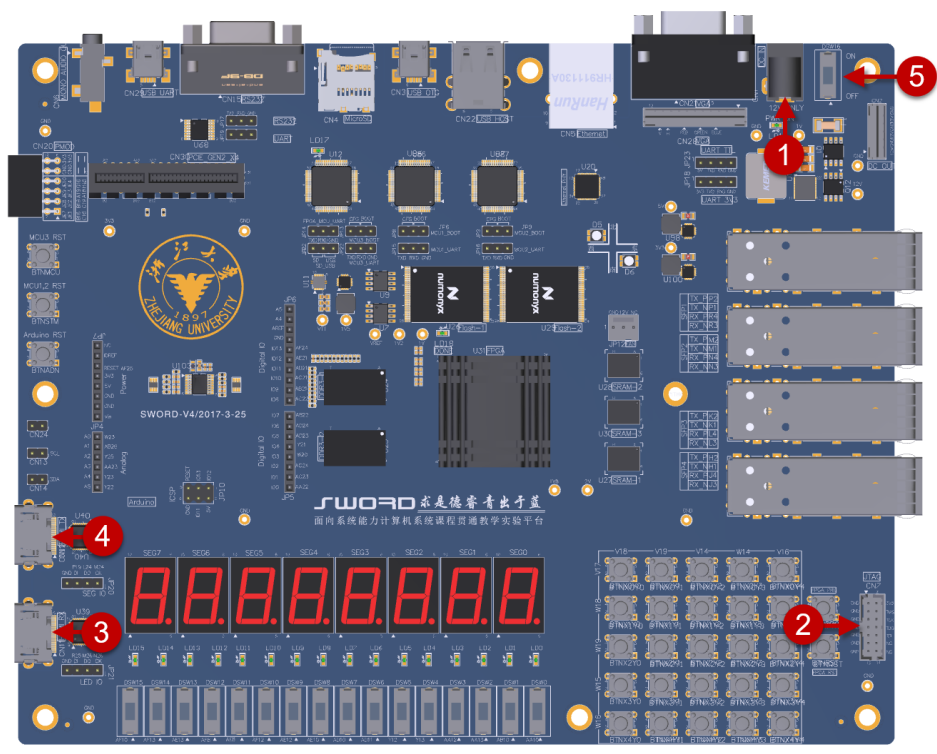



图 2-33 硬件连接对应位置

连接好后的效果如下图所示：



图 2-34 实际硬件连接

- 15 接着在 Hardware Manager 界面下，点击 Open target，在随之弹出的菜单中选择 Auto Connect，整个过程如下图所示：

	标题	文档编号	版本	页
	Lab7: 图像处理滤波器实验 4	XDI-SWORD-IMG-007	1.0	28 of 31
	作者	修改日期	公开	
Joseph Xu		2018/5/13		

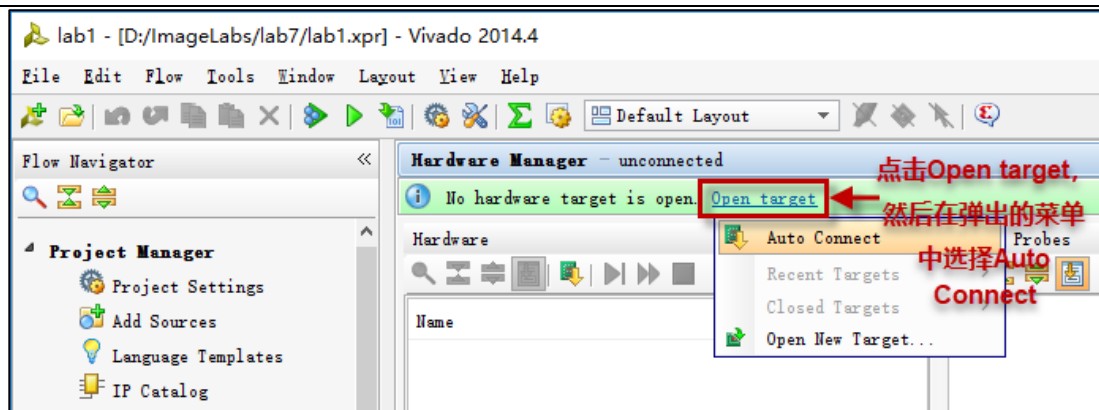


图 2-35 Open target

接着 Hardware Manager 会自动连接下载器并扫描 JTAG，一切正常的话，会显示出扫描到的目标器件：xc7k325t，鼠标右键单击目标器件，在弹出的窗口中选择 Program Device，整个过程如下图所示：

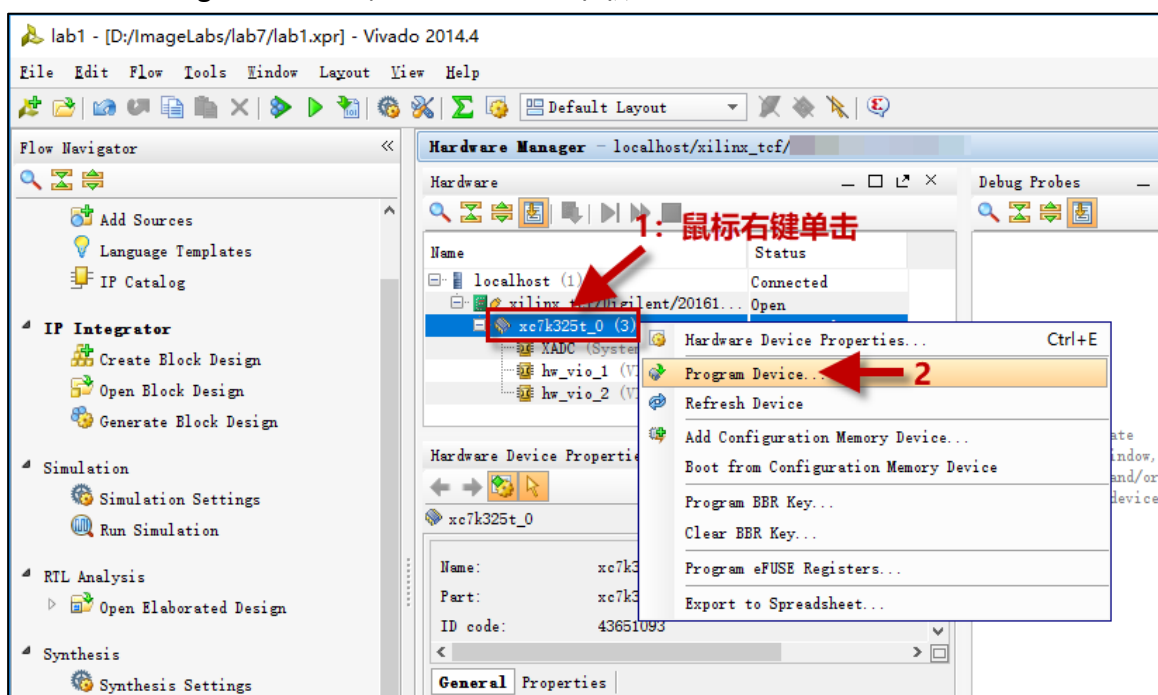



图 2-36 Program Device

在弹出的对话框中，保持默认设置，直接点击 Program，如下图所示：

提示：如果 Debug probe file 这一栏有输入，可忽略之。

	标题	文档编号	版本	页
	Lab7: 图像处理滤波器实验 4	XDI-SWORD-IMG-007	1.0	29 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/5/13		

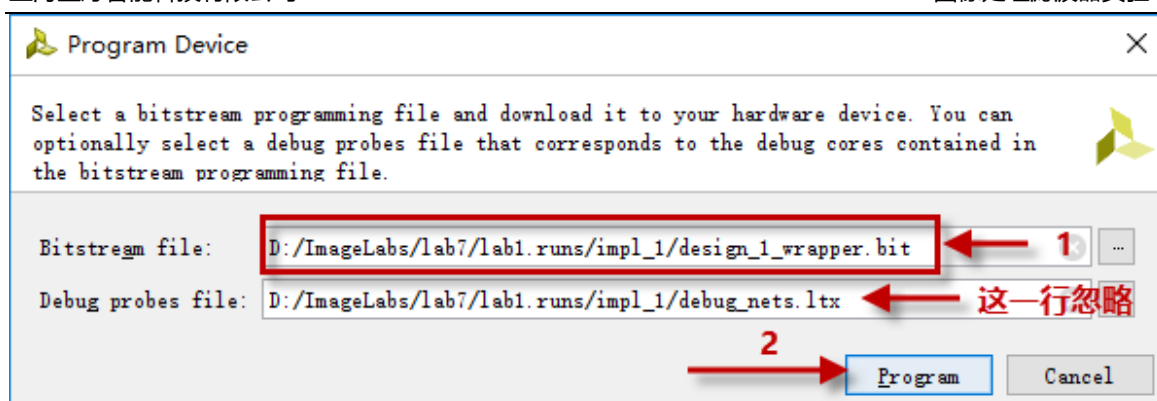


图 2-37 烧写目标器件

随着如下图所示进度条显示 100%，即表示目标器件烧写完毕。即可进入实验现象观察阶段。

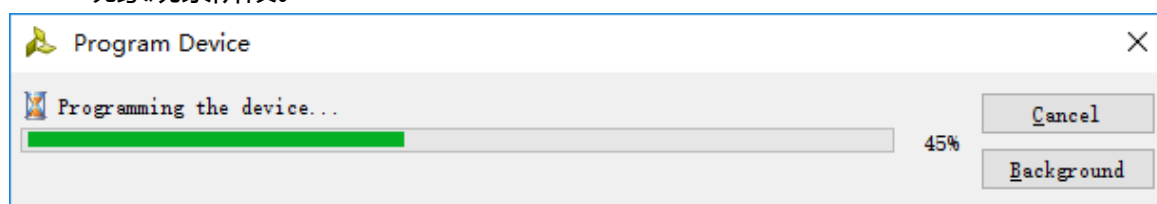



图 2-38 编程进度条

	标题	文档编号	版本	页
	Lab7: 图像处理滤波器实验 4	XDI-SWORD-IMG-007	1.0	30 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/5/13		

3. 二值腐蚀实验结果

我们将连接 HDMI 输入端口的 HDMI 线在信号源端重新插拔一次，以便让信号源设备重新检测 (Detect) 一下接收设备，一切正常的话，我们即可在 HDMI 显示器上看到经过局部阈值化滤波后的显示画面。

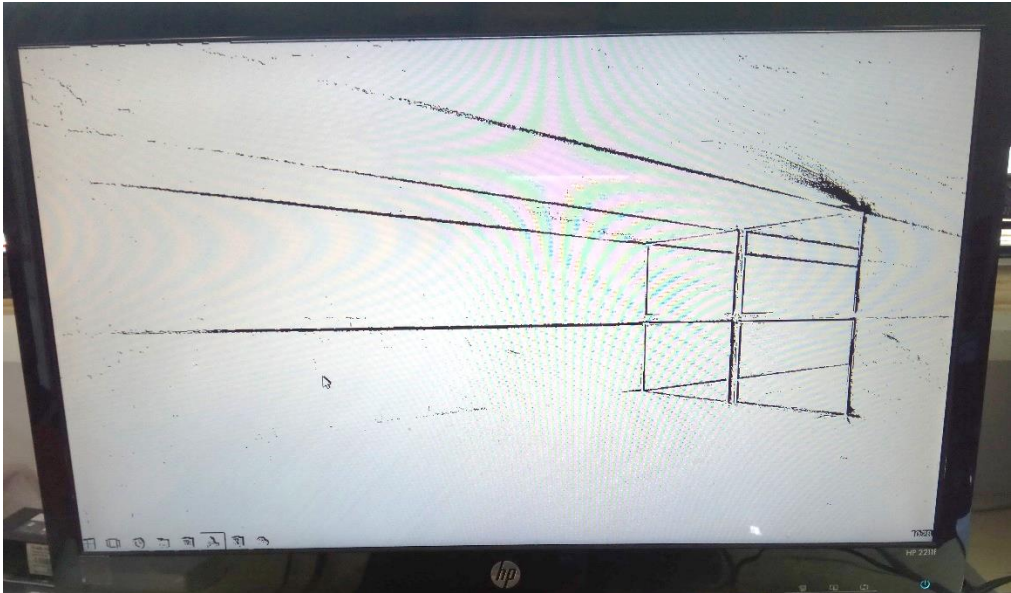



图 3-1 局部阈值化显示结果

	标题	文档编号	版本	页
	Lab7: 图像处理滤波器实验 4	XDI-SWORD-IMG-007	1.0	31 of 31
	作者	修改日期	公开	
	Joseph Xu	2018/5/13		